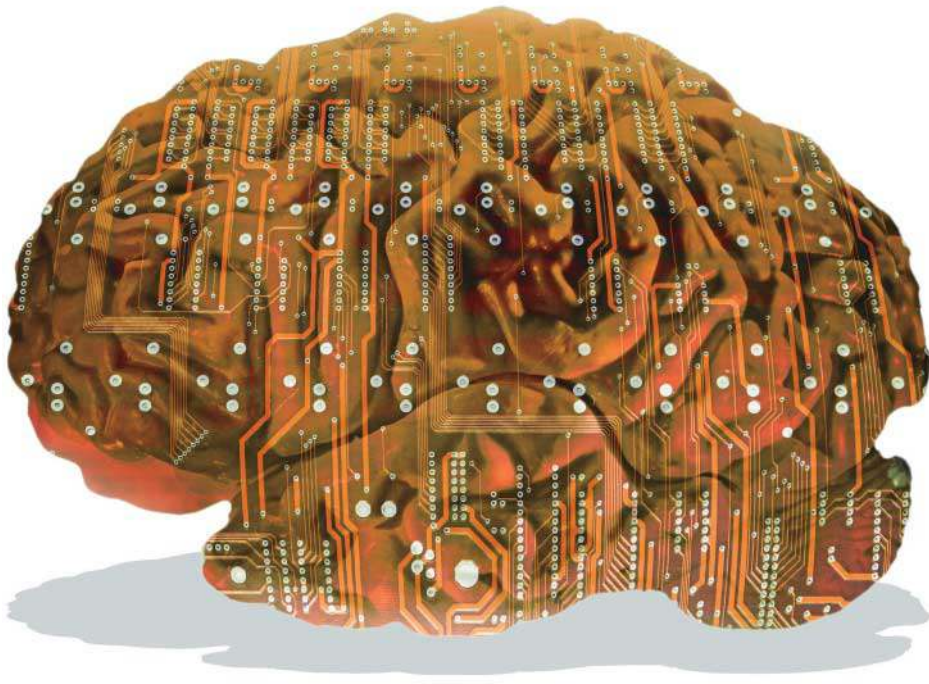


Was In-Memory-Datenbanken leisten

Im Gedächtnis

Mathias Golombek



RAM als Datenspeicher – auf dieses Prinzip setzen In-Memory-Datenbanksysteme (IMDB). Sie nutzen den Hauptspeicher eines Servers als beschleunigenden Cache und sind damit deutlich performanter als herkömmliche Datenbanken. Besonders im Big-Data-Bereich, wo große Datenmengen analytisch verarbeitet werden müssen, können IMDBs ihre Vorteile ausspielen.

Die Einsatzszenarien für IMDBs sind heutzutage so zahlreich, dass von einem Nischenprodukt keine Rede mehr sein kann. Neben Systemen, die konsequent auf die In-Memory-Technologie setzen, gibt es von fast allen großen Datenbank Anbietern – von Oracle bis IBM – ein In-Memory-Add-on für ihre sonst diskorientierten Systeme. Bei den verschiedenen Angeboten ist der In-Memory-Ansatz also unterschiedlich stark

ausgeprägt – mit den im Folgenden beschriebenen Vor- und Nachteilen.

Die Nachfrage nach einer IMDB entsteht immer aus der Praxis, vor allem dann, wenn Unternehmen ihre Geschäftsentscheidungen mehr und mehr anhand von Daten und deren Analyse treffen. Zwar gibt es IMDBs auch für transaktionale Systeme, die die Daten dort verarbeiten, wo sie entstehen, zum Beispiel auf Websites. Dafür sind jedoch

NewSQL-Lösungen wie MongoDB oder Couchbase zumeist die bessere Wahl. Statt mit klassischen Tabellenzeilen und -relationen arbeiten diese mit anderen Strukturen wie Key-Value-Stores. Solche Systeme ermöglichen eine besonders performante und skalierbare Verarbeitung einzelner Objekte.

Für analytische Anwendungsfälle ist jedoch die Abspeicherung in Relationen vorteilhafter. Denn für die Auswertung werden große Datenmengen miteinander verknüpft, etwa mittels Joins, Aggregationen und analytischen Funktionen. Das macht die IMDB besonders für diese Szenarien interessant – als Teil eines Data Warehouse und einer Business-Intelligence-Strategie.

Die In-Memory-Technologie bearbeitet Querys deutlich schneller als herkömmliche relationale Datenbanksysteme, weil sie den Hauptspeicher als beschleunigenden Cache nutzen kann. Klassische, relationale Datenbanksysteme müssen aufgrund des Flaschenhalses Diskzugriff immer große Blöcke von den Festplatten einlesen und verarbeiten diese dann nacheinander. Das ist in etwa so, als müsste ein Telefonbuch mit Tausenden Seiten nach dem Vornamen anstatt nach dem Nachnamen umsortiert werden und dabei dürfte immer nur eine Seite gleichzeitig geöffnet sein. Es könnte dann immer nur die eine Seite lokal sortiert und mit den beiden Nachbarseiten verglichen werden. Ein enormer Overhead entsteht, der die Verarbeitung extrem verlangsamt. Im Gegensatz hierzu ermöglichen IMDBs Random Access auf die Daten. Jede Speicherzeile kann dabei direkt über ihre Speicheradresse angesprochen werden, und zwar rasend schnell. Bei dem Beispiel des Telefonbuchs könnten alle Seiten offen liegen und auf jedes Element kann direkt zugegriffen werden. Insgesamt wäre die Umsortierung um viele Faktoren schneller.

Höher, schneller, weiter ...

In beinahe allen Industriezweigen sind datengetriebene Prozesse mittlerweile die Basis für Businessentscheidungen. Dabei geht es keineswegs nur um traditionelles Data Warehousing und das Erstellen von Reports mithilfe von BI-Tools. Der Trend geht immer mehr Richtung Near-Real-Time-Entscheidungen im ganzen Unternehmen, bei denen es auf eine möglichst niedrige Latenz ankommt. Je höher die Geschwindigkeit ist, mit der komplexe Anfragen ausgewertet werden, desto zügiger können Entscheider reagieren und

umso mehr Analysen lassen sich durchführen. Bei Anwendungen, die auf automatisierten Entscheidungsalgorithmen beruhen, wie etwa Dynamic Pricing oder IoT-basierte Laufzeitoptimierungen von Industrieanlagen, ist der hohe Durchsatz beim Analysieren ein wichtiger Wettbewerbsfaktor. Gerade in den Bereichen Industrie 4.0 (Predictive Maintenance), IoT (autonomes Fahren) und den neuen Geschäftsfeldern im Fintech-Umfeld kommt man ohne leistungsstarke Datenanalysen nicht mehr aus.

Das grundsätzliche Merkmal einer IMDB ist die Nutzung des Hauptspeichers als primäres Speichermedium. Dank moderater RAM-Preise fällt der Hauptspeicher heute nicht stärker ins Gewicht als andere Hardwarekomponenten. Zudem lässt sich in Clustern aus vielen günstigen Standardservern eine enorme Hauptspeichermenge vereinen und somit eine hohe Performance erzielen. Die benötigte Größe des Hauptspeichers hängt dabei natürlich zum einen von der Datenmenge ab. Zum anderen aber entscheidet die Art und Weise, wie stringent die In-Memory-Technologie eingesetzt wird, wie viel Hauptspeicher sinnvoll ist. Der Markt bietet im Wesentlichen drei unterschiedliche Ansätze: Systeme, die ausschließlich den Hauptspeicher für die Datenverarbeitung und -speicherung nutzen, solche, die nur die sogenannten Hot Data in den Hauptspeicher laden und verarbeiten, und schließlich klassische DBMS, die In-Memory als Add-on-Option anbieten.

Die erste Variante – alles im RAM – bedeutet die konsequenteste Nutzung der In-Memory-Technologie und damit der Performancevorteile des Hauptspeichers: Alle Daten stehen hier permanent zur Verfügung, zu keiner Zeit muss das System aus einem Disksystem etwas nachladen. Folgerichtig muss der Hauptspeicher entsprechend dimensioniert sein – je größer die zu verarbeitende Datenmenge, desto mehr Hauptspeicher ist notwendig. In dem Moment, in dem die Daten nicht mehr in den Hauptspeicher passen, muss

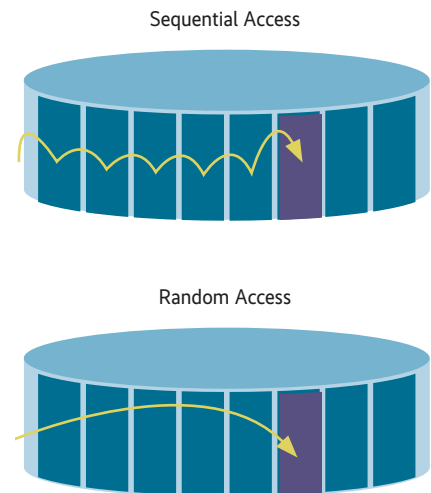
nachgerüstet werden. So müssen Administratoren einen relativ großen Hardwarepuffer einplanen, auch wenn die Datenbank nur temporäre Spitzenlasten abdecken muss. Das wirkt sich natürlich auf die Hardwarekosten aus. Prominenter Vertreter dieser Technologie ist SAP HANA.

Platten für die Persistenz

Da RAM ein flüchtiger Speicher ist, sind entsprechende Vorsichtsmaßnahmen zu treffen, um die Persistenz der Daten zu gewährleisten. Deshalb müssen auch In-Memory-Datenbanken alle Datenänderungen auf Festplatte schreiben. In der Praxis bedeutet dies jedoch keine wirkliche Einschränkung, da bei analytischen Anwendungen das Auswerten der Daten im Vordergrund steht und nicht ein möglichst schnelles Ändern einzelner Datensätze, wie es bei transaktionalen Systemen der Fall ist. IMDBs können ihre Daten im Hauptspeicher sofort ändern und dann im Hintergrund auf die Platte schreiben, während gleichzeitig die Analysen weiterhin auf den schnellen Hauptspeicher zugreifen können.

Während die Gesamtperformance solcher Alles-im-RAM-Systeme also sehr hoch ist, zahlt man dafür einen Preis, denn die Menge an Hauptspeicher, die eine solche IMDB erfordert, ist unter Umständen enorm. Aufgrund paralleler Analyseanfragen vieler Nutzer und temporärer Zwischenergebnisse für die Berechnungen sind meistens deutlich mehr als 100 Prozent der eigentlichen Datenmenge vonnöten. Für ein 10-TByte-Datenbanksystem sollte man somit 15 bis 20 TByte RAM bereitstellen.

Die zweite IMDB-Variante ist die Hauptspeicher-optimierte Datenverarbeitung mit dynamischem Caching. Diese Systeme, wie beispielsweise die In-Memory-Datenbank von Exasol, können ihre Daten sowohl im RAM als auch auf Festplatten speichern. Im Hauptspeicher



Sequenzielle Datenverarbeitung und Random Access im Vergleich

solcher Hybridsysteme befinden sich die jeweils „heißen“ Daten (Hot Data), also solche, über die gerade Querys, Joins oder andere Bearbeitungsschritte laufen. In der Praxis sind nur etwa 30 bis 50 Prozent der Rohdaten „heiß“. Durch die Kompression der Daten verringert sich die Datenmenge nochmals.

Gegenüber dem zuvor beschriebenen Full-In-Memory-Ansatz müssen Hybridsysteme regelmäßig Daten von Platte nachladen. Dieser Scan der Daten ist jedoch weit weniger rechenintensiv als komplexe Joins, Sortierungen oder Aggregationen. Kompressionsalgorithmen, Workload-Management-Systeme und weitere Tools wie Query Optimizer können den Overhead noch weiter reduzieren. Im Schnitt werden lediglich zehn Prozent Hauptspeicher im Vergleich zur Menge der Daten benötigt, bei gleichzeitig fast optimaler Performance.

Sollte sich während der Verarbeitung der Hauptspeicher als zu klein erweisen, sinkt die Performance eines Hybridsystems jedoch entsprechend, weil analog zu einem Betriebssystem-Cache immer mehr Daten hinein- und herausgeswappt werden müssen.

Müssen dauerhaft mehr Daten verarbeitet werden, greifen mehr Nutzer auf das System zu oder erhöht sich die Komplexität der Abfragen, so analysiert das Workload-Management-System die genaue Speichernutzung und meldet dann, ob und wie viel Hauptspeicher oder Serverknoten nachgerüstet werden sollten. Intelligente Query Optimizer legen für den Nutzer transparente Indizes an oder replizieren kleinere Tabellen, um die Performance zu erhöhen und den Hauptspeicherbedarf möglichst klein zu halten,



- In-Memory-Datenbanken spielen ihre Stärken im Bereich der schnellen Datenanalyse aus.
- Bei der Umsetzung existieren verschiedene Ansätze: von „alles im RAM“ über Hybridsysteme bis zu In-Memory-Add-ons für herkömmliche Datenbanken.
- Um den womöglich hohen Migrationskosten gerecht zu werden, sollte eine IMDB eine möglichst offene und erweiterbare Architektur haben.

ohne zusätzliche Tuning-Aufwände zu verursachen. Je intelligenter das System arbeitet, desto unsichtbarer wird die In-Memory-Technologie für den Nutzer.

In-Memory als Add-on für klassische Datenbanken

Fast alle etablierten Datenbankanbieter – von Oracle bis IBM und Microsoft – haben in den letzten Jahren für ihre Systeme ein In-Memory-Add-on entwickelt. Zwar beschleunigt bei solchen Erweiterungen der Hauptspeicher die Verarbeitung ebenfalls, einen grundsätzlichen Paradigmenwechsel in der Art der Datenverarbeitung bedeuten sie jedoch nicht. Sie verarbeiten weiterhin Blöcke, die nun im RAM gecacht werden, die Algorithmen hingegen bleiben gleich. Echter Random-Zugriff ist nicht möglich, da die Software dafür nicht ausgelegt ist. Deswegen sind zahlreiche Workarounds notwendig, um die In-Memory-Technologie zu integrieren.

Ob ein In-Memory-Add-on ausreicht oder die Erweiterung oder Ablösung durch eine „echte“ IMDB sinnvoll ist, muss der konkrete Anwendungsfall entscheiden. Ein großer Vorteil der Add-ons ist, dass sich bestehende Datenbanksysteme ohne tief greifende Änderungen beschleunigen lassen. Es fallen zwar zusätzliche Hardware- und Lizenzkosten an, aber eine womöglich aufwendige Migration bleibt dem Unternehmen so erspart. Allerdings darf man sich von den zu erwartenden Leistungszuwächsen nicht zu viel erhoffen. Gerade wenn das bestehende System deutlich mehr Datenvolumen verarbeiten muss, neue Applikationen dazukommen oder viel mehr Nutzer als bisher Zugriff auf die Datenbank erhalten sollen, lohnt sich die Evaluierung einer nativen IMDB. In vielen Fällen kann es auch sinnvoll sein, eine IMDB neben ein existierendes, konventionelles DB-System zu stellen, um für die wichtigsten Geschäftsanwendungen die höchste Performance zu gewährleisten. Andere Applikationen migriert man dann nach und nach, um den Umstieg so reibungslos wie möglich zu gestalten.

MPP, Shared Nothing und NVRAM

Die Rechenkapazität und die Speicherausstattung eines einzelnen Servers sind begrenzt. Moderne Datenbanksysteme nutzen deshalb immer öfter Massively Parallel Processing (MPP). Gerade

In-Memory-Datenbanken profitieren von dieser Technik, da sich durch das Zusammenschalten von Servern zu Clustern die Menge an verfügbarem RAM fast grenzenlos erweitern lässt.

Während traditionelle Shared-Disk-Architekturen vor allem bei Datenbank-Anwendungen aufgrund der notwendigen Synchronisierung der Lock Table und Buffer Pools Skalierungsprobleme haben, steckt hinter MPP eine Shared-Nothing-Architektur. Dabei sind die Server über das Netzwerk verbunden und teilen sich Bandbreite, jeder Knoten führt seine Aufgaben jedoch unabhängig mit seinen eigenen Kapazitäten aus und speichert „seine“ Daten auf den lokalen Disks. Ein Hash-Algorithmus verteilt die Daten gleichmäßig auf die zur Verfügung stehenden CPU-Kerne, sodass der Nutzer sich nicht darum kümmern muss.

Um die Skalierbarkeit noch zu erhöhen und 100 oder mehr Server zu verbinden, nutzen einige IMDB-Anbieter das SPMD-Programmiermodell (Single Program, Multiple Data). Es ist die konsequenteste Art, Daten parallel zu verarbeiten, denn dabei läuft auf jedem Server das gleiche Programm mit unterschiedlichen Daten. Die Server werden zu SIMD-Einheiten (Single Instruction, Multiple Data): Alle erhalten den gleichen Befehl und wenden diesen auf ihre lokalen Daten an. Durch die Vermeidung eines zentralen Master-Knotens, der die Arbeit verteilt, bietet das SPMD-Modell eine sehr hohe lineare Skalierbarkeit.

MPP auf der Basis von SPMD stellt die Performance auch für die Verarbeitung großer Datenmengen sicher. Zudem beschleunigt es den Scanprozess, wenn bei IMDB-Systemen, die nicht die gesamte Datenmenge im Hauptspeicher halten, Daten aus dem Festplattenspeicher nachgeladen werden müssen. Denn die Summe aus dutzenden Disks pro Server und einer Vielzahl günstiger Standardserver führt zu einer sehr hohen Scangeschwindigkeit.

Soll Hochverfügbarkeit im Cluster sichergestellt werden, lässt sich dies durch Replizierung über die Serverknoten hinweg erreichen. Fällt ein Server aus, springen sofort Ersatzserver ein, ähnlich dem Vorgehen bei einer RAID-Spiegelung. Kritische Systeme werden zusätzlich über zwei Rechenzentren hinweg gespiegelt. Welche Maßnahmen hier notwendig und sinnvoll sind, entscheidet der konkrete Anwendungsfall.

Nicht nur durch MPP, sondern auch durch schnelleren RAM soll sich künftig die Leistungsfähigkeit von IMDBs noch erhöhen lassen. Eine neuere Entwicklung

in dieser Richtung ist NVRAM (Non-volatile RAM). Es gilt als Fortführung des SSD-Ansatzes, also des noch schnelleren persistenten Speicherns von Daten, in diesem Fall noch näher an der CPU. Auf den eigentlichen Rechenprozess einer analytischen IMDB hätte der Einsatz dieser Technologie derzeit zwar keine besonderen Auswirkungen, denn der Lesezugriff ist sogar etwas langsamer als bei normalem RAM. Wohl aber kann NVRAM bewirken, dass die Daten schneller persistent gespeichert werden. Beim aktuellen Stand der Dinge kann NVRAM höchstens als zusätzliche Zwischen-Persistenzschicht fungieren, denn für den durchgängigen Ersatz herkömmlicher Festplatten oder SSDs ist die NVRAM-Technologie noch viel zu teuer.

Cloud-tauglich und anschlussfreudig

Die Anforderungen an Datenbanken haben sich in den letzten Jahren deutlich gewandelt. Nicht nur sind die Datenmengen gewachsen, vielmehr wird die Auswertung und Interpretation der Daten zunehmend zur Basis für Geschäftsentscheidungen. Das Feld der Data Science – also des Wissensgewinns aus Daten – entwickelt sich rasant, Anwendungen, Konnektoren und Architekturen, die wiederum zahlreiche Programmiersprachen verwenden, wollen integriert werden.

Jedes ernst zu nehmende System verfügt daher über Standardschnittstellen wie ODBC, JDBC, .NET und SQL zur Anbindung von Datenintegrations-Tools (Upstream) und Business-Intelligence-Anwendungen (Downstream). Vorteilhaft ist es, wenn ein IMDB-System zudem offene, möglichst abstrakt gehaltene APIs bietet. Durch diese Abstraktionsschicht ist es zukunfts offen und lässt sich flexibel erweitern, ohne dass der Hersteller dafür eine neue Schnittstelle in einer neuen Release ausrollen muss.

Zu einem immer wichtigeren Entscheidungskriterium bei der Auswahl wird, dass eine In-Memory-Datenbank cloud-ready ist. Ob jedoch eine Cloud generell die passende Infrastruktur ist, kommt auf die Ziele des Unternehmens an.

Zum einen gibt es Dienstleister, die die Datenbank und die Infrastruktur für ein Unternehmen auf dedizierten Servern als Software as a Service (SaaS) zur Verfügung stellen. Bei solchen gehosteten Private-Cloud-Angeboten geht es in erster Linie darum, Infrastruktur nicht selbst anschaffen zu müssen und auf die Update- und Support-Dienstleistungen des

Cloud-Dienstleisters zurückgreifen zu können.

Die als Hauptvorteil der Public Cloud angepriesene Flexibilität bei der Skalierung ist für Datenbanken meist gar nicht so relevant. Vor allem beim Einsatz für High-End-Analytics für geschäftskritische Prozesse laufen die Systeme dauernd unter Vollast. Ein agiles Up- und Downscaling würde durch die ständige Datenreorganisation und das Auffüllen des RAM-Cache den Performancevorteil von In-Memory effektiv zunichtemachen. Und zumeist spart man durch diesen Ansatz der flexiblen Ressourcen in der Cloud gerade bei High-Performance-Anwendungen gar kein Geld, wenn man ihn mit einem statischen, aber dedizierten und leistungsoptimierten Hardware-Setup vergleicht.

Als Entscheider muss man sich zudem stets die Frage stellen, wie abhängig man sich durch den Weg in die Cloud von bestimmten Diensten macht. Ideal ist es daher, wenn Datenbanksysteme sowohl den Betrieb on Premises als auch in der Cloud ermöglichen, am besten sogar parallel und auf verschiedenen Plattformen wie Amazon Web Services, Microsoft Azure und der Google Cloud Platform. So behält man maximale Flexibilität für die Firmenstrategie der Zukunft.

Auswahlkriterien für IMDB-Systeme

Bei der Entscheidung für oder gegen ein IMDB-System sollte man genau abwägen, was man braucht. Das Hauptkriteri-

um wird fast immer die erwartete Leistungsfähigkeit sein. Die Performance hängt jedoch von der verwendeten In-Memory-Variante, der konkreten Datenmenge, der Komplexität der Analysen und zahlreichen anderen Faktoren ab. Am besten definiert man für sich einen eigenen Performance-Benchmark und testet mehrere Anbieter unter einem realen Workload, um den für den speziellen Fall optimalen Anbieter zu finden.

Mindestens ebenso wichtig im Auswahlprozess sind die Integrierbarkeit und die Erweiterbarkeit der IMDB. Die Entwicklung von Unternehmen hin zu datengetriebenen Ökosystemen hat gerade erst begonnen, ein IMDB-System sollte deshalb so offen wie möglich sein – nicht nur für bestehende Technologien und Sprachen, sondern auch für künftig relevante. Die Kompatibilität zu ETL-Tools und BI-Anwendungen ist dabei ebenso wünschenswert wie die Einbindung von Big-Data-Software wie Hadoop, Spark und Kafka. Für den Einsatz im Bereich Data Science und KI sollte die Datenbank am besten gleichzeitig R, Python, TensorFlow und Java unterstützen oder sogar ein Framework, in das sich beliebige Sprachen integrieren lassen.


Weiterhin gilt es, bereits im Vorfeld zu bedenken, dass die IMDB die Basis für sensible und geschäftskritische Datenanalysen sein wird. Der ACID-Kompatibilität (Atomicity, Consistency, Isolation, Durability) kommt somit eine hohe Bedeutung zu. Während im operationellen Bereich zum Teil darauf verzichtet werden kann, ist deren Einhaltung im analytischen Anwendungsbereich essenziell.

Typische Beispiele sind die Analyse von Finanzzahlen und Umsatzdaten. ACID-Kompatibilität ist bei analytischen IMDBs deshalb genauso Grundvoraussetzung wie bei traditionellen relationalen Datenbanken.

Fazit

Einen umfassenden Eindruck davon, ob eine IMDB ins eigene Unternehmen und zur gewählten Infrastrukturstrategie – von on Premises über Public und Private Cloud bis hin zu hybriden Ansätzen – passt, verschafft ein Proof of Concept. Damit kann der Performancegewinn für die konkrete Anwendung geprüft und gleichzeitig das Optimum aus Datenmenge, Speicherdimension und Kosten austariert werden. Zudem empfiehlt es sich, IMDB-Projekte klein zu starten: mit der Beschleunigung einer bestimmten Analyse beispielsweise, um Kunden einen neuen Service zu bieten oder weil das Management fundierte Entscheidungen auf aktueller Datenbasis treffen möchte. Solch ein moderater Start lässt die Technologie langsam in das Unternehmen hineinwachsen, vermeidet hohe Migrations- und Integrationsaufwände und liefert schnelle Ergebnisse. (akl@ix.de)

Mathias Golombek

ist Vorstandsmitglied und CTO der Exasol AG. An der Universität Würzburg hat er Informatik mit den Schwerpunkten Datenbanken und verteilte Systeme studiert. 

Anzeige